Querying Wheatgenetics Database from R

June 14, 2017

Database Connection and Query Code

This shows how to connect to the K-State wheatgenetics database and retrieve phenotypic and genotypic information to be used in analysis. This requires RMySQL and the tcltk package.

Function to Get Login Details

This function creates a dialog box to obtain the user name and password. If an error occurs, this function can be bypassed as shown in the code below. This is a convenience function.

```r
#Login details for database that hide password and username
getLoginDetails <- function()
{
  ## Code by Barry Rowlingson
  ## http://r.789695.n4.nabble.com/tkentry-that-exits-after-RETURN-tt854721.html#none
  require(tcltk)
  tt <- tktoplevel()
  tkwm.title(tt, "Get login details")
  Name <- tclVar("Login ID")
  Password <- tclVar("Password")
  entry.Name <- tkentry(tt, width="20", textvariable=Name)
  entry.Password <- tkentry(tt, width="20", show="*",
                            textvariable=Password)
  tkgrid(tklabel(tt, text="Please enter your login details."))
  tkgrid(entry.Name)
  tkgrid(entry.Password)
  OnOK <- function()
  {
    tkdestroy(tt)
  }
  OK.but <- tkbutton(tt, text=" OK ", command=OnOK)
  tkbind(entry.Password, 
         "<Return>", OnOK)
  tkgrid(OK.but)
  tkfocus(tt)
  tkwait.window(tt)

  invisible(c(loginID=tclvalue(Name), password=tclvalue(Password)))
}
```

Connect to the Database

```r
log_detail=getLoginDetails() #fill in username and password in popup box

## Loading required package: tcltk
require(RMySQL) #load required packages may need to install.packages("RMySQL")
```

## Loading required package: RMySQL

## Loading required package: DBI

#dependencies are DBI package

#Make a MySQL connection to beocat
wheatgenetics=\dbConnect(\MySQL(),\texttt{user}=\texttt{log\_detail}[1],
\texttt{dbname}=\texttt{''wheatgenetics'', host='beocat.cis.ksu.edu',
\texttt{password}=\texttt{log\_detail}[2], port=6306) #run this line to connect to the database

#change dbname for the database of interest cimmyt or intermediate\_wheatgrass

######## If log\_detail does not run

######## Can provide login information as

######## Option 1

#wheatgenetics=dbConnect(MySQL(),user=\texttt{''your\_user\_name''},
\texttt{dbname}=\texttt{''wheatgenetics'', host='beocat.cis.ksu.edu',
\texttt{password}=\texttt{''your\_password'', port=6306}) #hard code in user name and password

#Note the password is saved in the code, and should be removed before distribution.

######## Option 2

#wheatgenetics=dbConnect(MySQL(),user=\texttt{''your\_user\_name''},
\texttt{dbname}=\texttt{''wheatgenetics'', host='beocat.cis.ksu.edu',
\texttt{password}=\texttt{\texttt{\textbackslash r}()}, port=6306) #hard code in user name

#Note the password is called by the \texttt{\textbackslash r}() function

#and the password is entered at the R console and pressing enter.

#If enter is not pressed the code will not execute any further.

###########Close Alternative Connection Options

#Check that the database connection is established
\dbListTables(wheatgenetics) #list all tables in wheatgenetics database

## [1] "Primary-synthetics\_summary" "barcodes"
## [3] "dna" "dnaQuant"
## [5] "experiment" "gbs"
## [7] "gbs\_iontorrent" "gcp"
## [9] "headrow" "htp"
## [11] "htp\_images" "htp\_instrument"
## [13] "htp\_phenotype" "htp\_phenotype\_run"
## [15] "htp\_traits" "kcia"
## [17] "ksu\_cimmyt\_plot\_map" "phemu\_htp"
## [19] "phemu\_images" "phemu\_run"
## [21] "phenocam\_images" "phenocam\_run"
## [23] "phenocorn\_htp" "phenocorn\_htp\_gis"
## [25] "phenocorn\_images" "phenocorn\_run"
## [27] "phenotype" "plant"
## [29] "plot" "plot\_map"
## [31] "seed" "seedinv"
## [33] "seedlocations" "sgbs"
## [35] "spiked\_locus" "tissue"
## [37] "tissueinv" "traits"
## [39] "trays" "uas\_images"
Notes

The sample code connects to wheatgenetics. If you wish to connect to another database such as the intermediate_wheatgrass or cimmyt database, the database name must be changed accordingly.

GBS Key File Query

Create a GBS key file query for Family 8 population. Example from Wheatgenetics Quick Start Guide.

```r
#make query statement
gbs_key_query = "SELECT gbs.flowcell Flowcell,
gbs.lane Lane,
barcodes.barcode Barcode,
dna.sample_name FullSampleName,
dna.plate_id,
substring(dna.well_A01,1,1),
substring(dna.well_A01,2,2),
concat(gbs.gbs_id,barcodes.barcode),
dna.well_A01,
dna.notes,
gbs.plexing,
gbs.project,
gbs.enzyme,
dna.sample_id,
dna.tissue_id,
dna.external_id,
dna.dna_person,
dna.line_num,
gbs.gbs_name,
dna.plate_name,
dna.well_01A,
plant.plant_name,
gbs.gbs_id,
barcodes.set"
FROM dna LEFT JOIN gbs ON gbs.dna_id = dna.plate_id
LEFT JOIN plant ON dna.tissue_id = plant.plant_id
INNER JOIN barcodes ON dna.well_A01 = barcodes.well_A01 AND
gbs.plexing LIKE barcodes.`set` = 1
WHERE gbs.project LIKE '%crain%' and gbs.`global` = 1
ORDER BY gbs.gbs_id, dna.well_01A ASC"

key = dbGetQuery(wheatgenetics, gbs_key_query) #get key from query

#return 1536 records
nrow(key)
```

## [1] 1536
Phenotypic Data Query

Query all phenotype information from the 2015_BYD_Experiment. Example from Wheatgenetics Quick Start Guide.

```r
# make query statement
phenotype_query="Select phenotype.*
From phenotype
Where phenotype.entity_id in (select plot.plot_id from plot where
plot.experiment_id = '15_ASH_BYD')"

phenotype=dbGetQuery(wheatgenetics, phenotype_query) # run query

# return 1654 observations
nrow(phenotype)

## [1] 1654

Final Steps

dbDisconnect(wheatgenetics) # disconnect from database

## [1] TRUE

Session Information

```r
sessionInfo()
```

## R version 3.3.3 (2017-03-06)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: macOS Sierra 10.12.5
##
## locale:
##
## attached base packages:
## [1] tcltk stats graphics grDevices utils datasets methods
## [8] base
##
## other attached packages:
## [1] RMySQL_0.10.11 DBI_0.6-1
##
## loaded via a namespace (and not attached):
## [1] backports_1.1.0 magrittr_1.5 rprojroot_1.2 tools_3.3.3
## [5] htmltools_0.3.6 yaml_2.1.14 Rcpp_0.12.11 stringi_1.1.5
## [9] markdown_1.5 knitr_1.16 stringr_1.2.0 digest_0.6.12
## [13] evaluate_0.10
```

4